Exhibit B

APPGATE®

Brainspace

Continuous Multimodal Learning

WHITEPAPER

Table of Contents

Executive Summary

Stay Tuned

This whitepaper on CMML and next-generation TAR workflow is the first of three whitepapers on TAR in Brainspace 6. Look for forthcoming whitepapers on our Predictive Coding approach to first-generation TAR workflows, and on the supervised learning technologies behind both CMML and Predictive Coding.

First-generation TAR (Technology Assisted Review) workflows, sometimes called TAR 1.0, feature supervised machine learning, iterative training on large batches of documents, and statistical evaluation of predictive model effectiveness. These workflows have substantially cut costs for large scale document reviews in litigation, second requests, and related applications. TAR software is now being applied to new tasks in litigation, including analysis of inbound productions and enhanced approaches to ECA (early case assessment). Applications outside litigation have emerged, including corporate investigations, information governance, law enforcement, and national intelligence. These next-generation applications demand dynamic and interactive workflows that exploit multiple text analytics tools. Many document reviews could benefit from new workflows, which have been promoted under names such as TAR 2.0, TAR 3.0, and others. Brainspace 6 introduces Continuous Multimodal Learning (CMML), a text analytics framework to support these next-generation workflows. CMML allows you to pursue lines of inquiry using a range of text analytics tools, and leverage the information discovered across all of them to drive training of predictive models. The result is fast-moving workflows where machine learning accelerates discovery rather than getting in the way.

First Generation of TAR Workflows

TAR software includes a range of text analytics tools, text classification, statistical ranked retrieval (in the guise of concept search in e-discovery), clustering, duplicate and near-duplicate detection, email threading, and communications analysis. Text classification, often referred to as predictive coding in e-discovery, is based on supervised machine learning. In supervised learning, you teach the software by manually coding documents, while a machine learning algorithm produces a predictive model from the coded data. The predictive model then assigns a predictive score to each document indicating its likelihood of belonging to a category of interest. The goal of first-generation TAR workflows, or what some have called TAR 1.0, has been to bring efficiencies to large document reviews traditionally carried out by rooms full of attorneys. Typical examples include responding to production requests in litigation and to second requests in antitrust investigations. Law firms, legal service providers, and corporations have developed a range of first-generation TAR workflows. While varying, these workflows tend to share some or all of the characteristics shown in the first column of Table 1. First-generation TAR workflows have focused on text classification as a tool for either reducing the volume of material to review (culling) or ensuring the most promising material is reviewed first (prioritization). Random samples have commonly been used to estimate the effectiveness of culling, using measures such as recall (the proportion of relevant documents that survive after culling). First-generation TAR workflows, as supported by products like Brainspace's market-leading text analytics software, have shown their ability to reduce the costs of large scale reviews. Results from statistical sampling have helped convince skeptical courts and litigation adversaries that these cost savings can be achieved while finding as much or more responsive material as traditional manual workflows.

1st Gen TAR Workflows Don't Meet Needs

Despite their successes, first-generation TAR workflows are not always a good match for emerging text analytics application areas such as corporate investigations, information governance, law enforcement, and national intelligence. Nor are they the best fit for all litigation applications. Tasks such as ECA (Early Case Assessment) and analyzing inbound productions put an emphasis on getting to key information fast rather than finding all documents of a particular type. And while large scale reviews in e-discovery are the canonical application for first-generation TAR workflows, factors such as rolling collections and productions, low richness, and time pressures pose challenges to first-generation TAR even there.

Traditional TAR Workflow Tools

Some downsides of first-generation TAR Workflows for next-generation applications are summarized in Table 1 below.

Tool	1st Gen Workflow	Downsides in Nex Gen Tasks
Control Sets	The TAR software draws a control set (a large random sample) at the start of training, and users manually code it. The control set is used to estimate richness, track the effectiveness of the evolving predictive model, and choose a cutoff for a classifier.	Manually coding a control set can be too expensive when richness is low or a project is small, too slow in a fastmoving investigation, and too static in the face of an evolving definition of relevance. Statistical estimates of predictive model effectiveness are not needed in all tasks.
Batching	Large batches of training examples, usually selected by the TAR software, are used in training. This is a natural fit for integrating supervised learning with traditional document review teams and workflows.	Batching may make less sense for a small team using an exploratory workflow. Once a few key documents are found, the project may end, or the topics of interest may change substantially, making previously labeled training batches of little value
Separating Training & Review	One team labels training and evaluation data, while a second team reviews documents found by the predictive model. This allows experts with limited availability to train the model, while a large team handles the mass of details of review.	Separating training and review makes less sense for an individual or small team whose goal is to find information, not process every document.

One of the most common complaints when applying first-generation TAR workflows in these faster-moving applications is that supervised learning is too cumbersome. Manual coding of large control sets and training batches, sometimes consisting largely of irrelevant documents, becomes an obstacle rather than an aid. Instead of quickly reaching key information, users spend too much time and energy training and evaluating the system. In addition, supervised learning, while powerful, is only one of many text analysis tools. Different tools find different types of documents and different types of information, and ideally are used together by a team for maximum effectiveness. Since next-generation applications focus on finding information rather than on processing every document, a workflow centered on batch training and classification of each document is not always the most effective approach.

Some Characteristics Typical of 1st Gen & Next Gen Workflows

1st Gen TAR Workflow	Next Gen TAR Workflow
Users train a predictive model and use it to cull or prioritize a document collection. Review of those documents happens sometime later	Users continuously train predictive models as documents of interest are found, and continuously use those models to find more documents of interest
One expert or a small team trains the predictive model. A different team reviews documents found by the model	A single team trains and uses predictive models to find relevant documents. They read those documents in support of a review, analysis or investigation
Training uses large batches of documents, often chosen by the TAR software itself	Training uses documents naturally found during a user search for information, regardless of how they were selected. User knowledge is combined with training data
A predictive model is trained until predictive effectiveness stops improving	Predictive models are trained when convenient, and only to the extent the provide value
Workflows are centered around supervised learning	Workflows integrated multiple analytics tools, any of which can feed supervised learning, and turn in, use its results
Evaluation focuses on the effectiveness of the predictive model	Evaluation focuses on the effectivenss of the overall workflow
Large random samples are used to estimate richness, track training progress, and set cutoffs for review	Multiple approaches are used for management and quality assurance. Random sampling is optional, and often reserved for a final validation check at the end of the process
The focus is on appropriate handling of every document	The focus is on finding the right information fast
Each project starts from scratch	Knowledge is accumulated across projects and applied to other similar sets of documents for instant effectiveness

Introducting Continuous Multimodal Learning (CMML)

Industry buzzwords such as TAR 2.0, TAR 3.0, Predictive Coding 4.0, and the like have sometimes brought more confusion than clarity. There are actually a range of new applications where text analytics is needed, and an even broader range of appropriate workflows. Table 2 presents some typical characteristics of these next-generation workflows, and contrasts them with first-generation workflows. However, just as with first generation workflows, not every next-generation workflow is same. Addressing next-generation workflows is therefore more complicated that just finding a single magic algorithm.

In response to next-generation applications, Brainspace has developed Continuous Multimodal Learning (CMML), an integrated set of technologies for supporting flexible, interactive workflows. These tools build on the foundation of Brainspace's established and market-leading text analytics suite. Given the diversity of workflows appropriate for next-generation tasks, CMML does not impose a single workflow model. Instead it provides a framework for carrying out a range of next-generation workflows, and text analytics tools for accomplishing those tasks and tracking progress.

In developing CMML, our goal was to make supervised learning a natural part of exploring a data set and seeking out key information, without requiring it to be at the center of the workflow. You can teach the system at your convenience and in the course of performing other forms of search, analysis, and review. Then apply the trained predictive models to find documents of interest, while at all times choosing the best analytics tool for each task, and pursuing lines of inquiry in the most natural fashion.

That sounds simple, but seamless integration of multiple tools is never easy under the hood. In developing CMML, Brainspace reexamined every aspect of TAR workflows and the software design that supports them.

CMML Innovations

Brainspace's CMML framework is based on a range of technical innovations to support creating work product, flexibly using supervised learning and other analytics, and managing projects.

Tools that allow your team to record their findings and integrate information across tools are critical for next-generation workflows. Brainspace CMML provides two powerful mechanisms for this: a tagging system integrated with supervised learning, and a notebook capability for collecting, combining, and reviewing sets of documents.

TAGGING

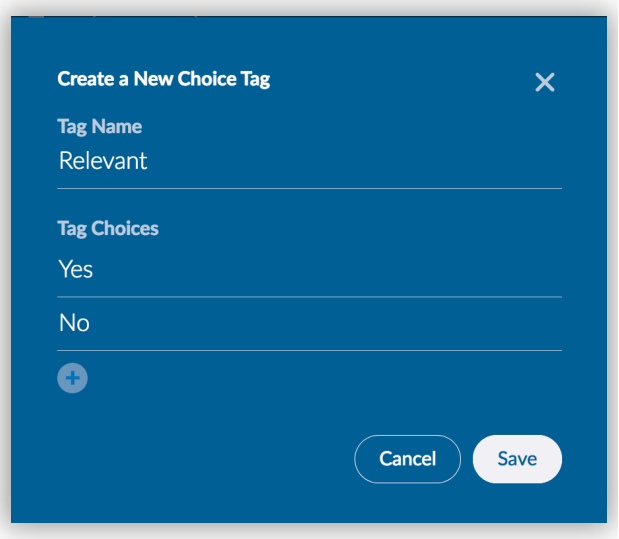
Brainspace 6 allows you to define tags of interest and link these tags to predictive models. Any time a document is viewed, it can be tagged for topics of interest. You can provide training data for multiple predictive models at once, without leaving a line of investigation. Then, at your convenience, apply supervised learning to the tagged documents to build a predictive model. Since you can define multiple tags and link them to multiple predictive models, you can easily experiment with a range of topics, and pursue or drop them according to the value they provide.

The notion of a "tag" has many meanings in information systems, from hash tags on the Internet to RFID chips in warehouses.

CHOICE TAGS

A choice tag is one that is defined to have a set of two or more values that may be chosen from a menu. Tagging may be slightly slower, but choice tags support several functionalities that single tags do not:

- Choice tags can be used to define negative examples for supervised learning.
- Choice tags can specify that a document is of interest, but is not to be used in supervised learning. Reasons to exclude a document (permanently or temporarily) from supervised learning include technical difficulties (encryption, bad OCR, foreign language, etc.), the need to get expert interpretation of the document, ambiguity, or because the document would make a poor training example.
- Choice tags can be used to verify that all documents in a certain set have been reviewed.



NOTEBOOKS

Even in an interactive and opportunistic workflow, you still need to work with sets of documents. Sets of results must be assembled, work must be coordinated among team members, and it may be necessary to ensure that all documents of a particular type have been reviewed. Notebooks in Brainspace 6 replace Brainspace Discovery's collections facility. Notebooks enable you to collect a set of documents for easy reference and querying. Add or delete individual documents to a notebook, or build up a notebook in larger increments using queries, clusters, predictive models, or other analytic tools. Tags may be associated with a notebook and tracked, thus supporting a lightweight review capability in Brainspace itself. Conversely, you can collect all documents with a particular tag into a notebook. The set of documents from a notebook can be viewed using Brainspace's pioneering cluster wheel visualization, and can serve as input to any of the other analytics tools. A notebook can serve as a training batch for supervised learning, so that results from all analytics tools can be used to drive machine learning.

SUPERVISED LEARNING

In first-generation TAR workflows, coding of training documents is done in much the same way that documents are coded in traditional manual reviews. Large batches of documents are selected (usually by the TAR software) and a series of batches are reviewed, with a model trained after each batch, before producing the final predictive model for use. This batch-oriented training process provided a familiar transition from batch oriented manual review workflows.

In next-generation applications, however, you may need to use predictive modeling at any point in your work, and to integrate its use smoothly with other analytics tools. CMML therefore allows predictive models to be trained and used in a much more flexible fashion.

TRAINING WHEN YOU WANT IT

Brainspace's tagging system makes document coding for supervised learning a byproduct of your work,

not an additional step to slog through. CMML accumulates documents as you tag them, and has them readily available when you want to use them to train a predictive model. A predictive model can be trained from as little as a single document of interest.

You can also easily search for and revise your past tagging decisions. This contrasts with some first-generation TAR approaches that make it difficult to change the contents of a training batch or the coding of training documents. The ease of updating tagging decisions eases worries of "getting it wrong" in supervised learning. For instance, you can tag documents broadly while the conception of a topic is still evolving, and then clean up that tagging when the issues at hand become clearer.

The lightweight nature of tagging, predictive model creation, and training reduces the investment to try out

new ideas. Tags for different topics, or different versions of the same topic can be defined. Coded documents from previous reviews, which may not be perfectly on point, can be used to kick off training, and then be removed or recoded later after more useful documents are found. You can take a notebook of documents, the results of a metadata search, or a cluster and use to train a throwaway predictive model to find related information.

Freed from the burden of heavyweight processes and fears of making a mistake, supervised learning reaches its potential as an extraordinarily powerful form of conceptual search: a More Like This on steroids.

PREDICTIVE RESULTS WHERE YOU WANT THEM

First-generation workflows tend to isolate predictive modeling from other aspects of text analytics. With first-generation TAR software, you can easily train a predictive model and use it to cull or prioritize a data set. But it can be awkward to examine top-scoring documents, train on them, or explore them using other analytics tools.

PREDICTIVE RESULTS WHERE YOU WANT THEM

In contrast, CMML closes the loop, providing full access to the results of supervised learning within the Brainspace application. You can rank documents using the score of any predictive model, and then quickly train a new model using top ranked documents. This technique is called iterative relevance feedback, and is a form of active learning often promoted for next-generation workflows (see sidebar). Top-scoring documents can also be batched for later review. Even better, results from predictive modeling can be analyzed using other analytics tools. A particularly powerful strategy is to use Brainspace's cluster wheel to create a focus on top-ranked documents from a predictive model. This combines supervised learning and unsupervised learning with a highly responsive exploratory visualization tool to explore the most promising portions of a large document collection and turn up new themes of interest.

Document scores can also be used in queries, just like any other metadata. This makes it easy to combine the results of predictive modeling with other criteria to rapidly zero in on the most important documents, and use them with any of Brainspace's analytics tools.

BATCHES IF YOU NEED THEM

Batching of documents may be less common in next-generation TAR workflows than in first-generation ones, but its uses are actually more varied. CMML provides the necessary support for these diverse uses of batches of documents. You can review training batches within Brainspace itself, using a special review mode that enables one-click tagging of training documents. Batches may also be developed and coded in an external review tool and synchronized with Brainspace. Tags can even be linked with fields in Relativity™. This allows Brainspace to serve as the core analytics function, while an organization's desired review platform is used for managing review and production. Predictive model scores can be synchronized back to Relativity as well.

Here are few examples of the ways that batches can be useful in workflows with CMML:

Powering Your Review Tool With CMML: You can manage a large-scale review in your existing review platform, and use the batches of documents coded there for supervised learning in Brainspace. CMML's predictive modeling can then be used to prioritize documents for the next round of review in the platform. This big batch style of next-generation review is a common alternative to first-generation workflows for large scale reviews.

Filling the Gaps in User-Selected Data: Concerns are sometimes expressed that user-selected training data will miss important subtopics, or even bias the predictive model. Brainspace's active learning algorithms choose batches of documents not only to accelerate training, but also to fill the gaps in existing training data in an anti-biasing fashion.

Standardizing Supervised Learning: Organizations sometimes develop checklists and workflows for supervised learning as a way to avoid oversights and build client confidence. Other organizations may be required to follow a process negotiated with adversaries in a litigation. A process might specify, for instance, that each of several analytics tools be used to provide a certain amount of training data. CMML's batching capability (and the progress tracking methods discussed in the next section) make it easy to implement and monitor such processes.

ALGORITHMS

CMML uses the same state-of-the-art supervised learning algorithm as Brainspace's Predictive

Coding (PC) framework, including our multilingual processing and concept formation. CMML additionally allows portable models to be used to kick off training, and metadata fields to be used as features in predictive models, providing additional power. All training batch selection methods, including our Fast and Diverse active learning algorithms, are available for both CMML and PC.

In addition, CMML supports selecting top-ranked untagged documents using the current predictive model and saving those as a notebook for training. This implements incremental relevance feedback, an active learning method invented by J. J. Rocchio in 1965, and recently popularized as "continuous" learning in e-discovery. Incremental relevance feedback is one of the many tools available in CMML for finding relevant documents in a low richness setting.

See our forthcoming white paper Text Classification: The Brainspace Approach for more on supervised learning in Brainspace 6.

TRACKING PROGRESS

In any workflow, managers must be able to track progress toward a goal. In first-generation TAR workflows, the focus is on creating and using a predictive model. Metrics and visualizations are focused on the effectiveness of the evolving predictive model. One example is the depth for recall graph in Brainspace's Predictive Coding capability.

In next-generation workflows, the goal may be to find most or all documents on a particular topic, much as in a first-generation workflow. However, instead of finding particular information, it might be used to demonstrate that particular information is not present. Different portions of a collection may also have very different review goals. Predictive models, while powerful, are just one of the tools used in accomplishing next-generation tasks. CMML therefore provides a broader range of progress tracking tools than do first-generation TAR systems. These tools include:

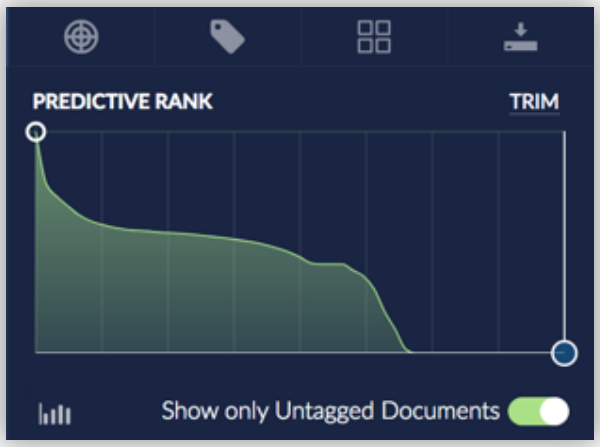
Notebook Review: Brainspace allows tags to be linked to a notebook, and tracks progress on reviewing the notebook for those tags.

Collection Review: Two tools support tracking the progress of finding positive examples for tags linked to a classification:

- CMML Progress Graph: This graph tracks the finding of positive documents for the classification against tagging effort.
- Validation Sets: You may draw a random sample at any time from all documents that have not been coded as positive or negative for a classification. After coding this sample, CMML will estimate the proportion of untagged documents that are positive examples. This may be done at the end of the project, or at any earlier point, with validation documents recycled for training if the CMML workflow continues.

Supervised Learning: Brainspace 6 supports metrics in both CMML and Predictive Coding that track the training of predictive models:

- Predictive Ranks: An interactive graph shows the distribution of scores from the predictive model, and allows navigating to particular score regions.
- Consistency: A measure of how much the predictive model agrees with the training data.
- Classifier Insights: Leverages our portable model capability to track how a predictive model is changing as training progresses.
- Stability: This measure in Brainspace reports indicates how much the predictive model has changed in response to recent training.



Not all next-generation workflows need all of these measures. For instance, some workflows are focused on finding a particular type of document within a collection, so the CMML Progress graph is particularly useful. Other workflows require that particular groups of documents have been reviewed, so using notebook progress indicators is helpful. The consistency and stability of the predictive model for a particular classification is useful in understanding the training progress of the model. Multiple analytics tools will typically operate in a CMML workflow, and often several predictive models as well, so progress on training one predictive model from that workflow is only a part of the story.

TARGETED LEARNING

Brainspace supports a powerful mechanism, focuses, for limiting analytics to particular documents of interest. A focus is subset of a data set, produced using any of Brainspace's analytics tools. A CMML classification can also be limited to a focus, so that training and use of supervised learning is limited to those documents. Restricting CMML to a focus has a range of purposes, including:

- Screening out documents known to be uninteresting to speed up training
- Deep dives into particular custodians, time periods, or other groups
- Separating documents by foreign language to build language specific models or aid tagging
- Separating documents by format (e.g. spreadsheets) for special handling

PORTABLE LEARNING

The first generation of TAR systems treats each project as starting with a blank slate, with a predictive model trained from scratch each time. Users increasingly demand the ability to leverage trained models across multiple projects, and to combine user knowledge with training data, and more generally to be instantly productive on new projects.

Brainspace's Portable Learning supports these needs. Predictive models trained on one data set can be exported as a portable model in a simple spreadsheet format. The portable model can be manually edited to remove features unlikely to be of interest on new data sets, or to add features based on user knowledge.

The portable model can then be imported into new data sets, providing immediate effectiveness. The imported models can then be tuned to the new data set using supervised learning. Predictive models can also be created from scratch, providing a way to immediately leverage user knowledge.

Portable models provide a way for Brainspace users to build valuable intellectual property that carries across projects and provides ongoing value.

Conclusion

Finding documents once meant keywords and complex query syntax, with all the limitations that brought. Concept search provided a great advance, but still requires effort to understand and balance vocabulary choices.

Brainspace's CMML is a new approach to finding documents, finding information, and finding resolutions. By making supervised learning as easy to use as search, you may bring its power to bear without specialized training or workflow overhead.

CMML provides the most flexible and efficient way to leverage supervised machine learning for investigations, litigation and intelligence mining.

CMML ENABLES YOU TO:

- Focus on the task, not the tool. CMML's supervised learning lets each new document you find accelerate the pace of locating the most crucial information.
- Reduce the time (and cost) to find key information. CMML augments your work without requiring large batches, sampling, or software-selected documents. Portable models allow instant productivity, by accumulating knowledge across projects.
- Tailor the workflow to the project. Your team can seamlessly combine the right tools for each project, tailoring them to your working style, and using them all to feed predictive modeling.
- Track progress and maintain quality. Tools for tracking progress and measuring quality including optional statistical sampling, support process requirements and management needs.

Brainspace CMML's support for next-generation workflows complements our Predictive Coding support for more traditional batch-oriented reviews. Look for our forthcoming white paper on Brainspace's innovations in Predictive Coding, including new visualizations and cost estimation features. Or contact us for a test drive of CMML, Predictive Coding, and the rest of our market-leading text analytic provides ongoing value.

The Brainspace Approach To Predictive Coding

William Webber, PhD.

William Webber Consulting
for Brainspace Corporation

1. Introduction

Predictive coding is the application of text classification to solve the problem of large-scale document production in electronic discovery. There are several predictive coding offerings on the market at present, but many of these embody questionable design decisions arising from legacy technology and poor modeling of the discovery for production task. In designing their predictive coding solution, Brainspace have been able to consider these design questions afresh. In this white paper, we give a general introduction to predictive coding, and discuss the distinctive features of the Brainspace discovery product.

2. Predictive Coding

A party to civil litigation in the United States and many other jurisdictions is faced with the obligation to produce material relevant to the case, and/or responsive to their counterparty's production requests. The traditional approach to performing this task was to have lawyers or paralegals read through all possibly relevant documents, page by page, and identify responsive material. With the exponential growth in electronically stored, unstructured corporate data, however, this approach of exhaustive manual review has long ago become infeasible. A stop-gap measure was to only consider documents that matched a keyword filter; but experience and experimental results have indicated that a keyword filter strict enough to reduce the manual review task to a manageable size will generally exclude too much responsive material—often as much as 80% of it. A variety of more advanced search technologies falling under the name of "concept search" have also been tried, but they have failed by themselves to offer the rigor and thoroughness required for use in civil litigation.

The most recent and successful approach to solving the task of document production in electronic discovery is that known as predictive coding. Under the hood, predictive coding uses a technology called statistical machine learning for text classification. You use text classifiers every day, though perhaps without realizing it: they form the central technology powering spam filters. The idea of statistical machine learning for text classification is that a computer program is provided with a large number of training example objects, labeled according to their class (usually by a human). From these examples, the program learns a model of what an object of a certain class looks like. Then, when shown a new and unlabeled target object, the program attempts to predict the class it belongs to, based upon the model it has learnt from its training data. In spam filtering, the objects are emails, and the classes are spam and ham (that is, valid emails). In predictive coding, the objects are documents, and the classes are responsive and non-responsive (plus potentially other issue codes such as "hot" or "privileged").

Text classification is a well-established technology. Predictive coding, however,

has many aspects to it that distinguish it from the standard text classification tasks one encounters in text books or the research literature. Indeed, in our view, there are two causes for the failings of some other predictive coding offerings: on the one hand, not breaking free of legacy technologies developed prior to predictive coding; but on the other, applying vanilla text classification techniques and processes, without properly adapting them to the peculiar features of electronic discovery. In the remainder of this white paper, we discuss the distinctive features of Brainspace's solution, and why we believe that these features provide the correct approach.

3. Predictive Coding in Brainspace Discovery

3.1 Logistic regression: powering predictions

There are a large variety of learning algorithms available for text classification: nearest neighbors, decision trees, naive Bayes, support vector machines, neural networks, and more. It can sometimes seem as if these technologies are interchangeable, and that the only question is which one gives the most accurate predictions. That impression, however, is misleading. There are several algorithms that give roughly the same, class- leading accuracy in prediction, but which differ on other features. A careful choice is required.

We identified the following five criteria for selecting a text classification algorithm for predictive coding:

- It must offer predictive accuracy that is at or close to the best across a range of published and internal experimental results. (This rules out naive Bayes and decision trees.)
- It must be quick to build its model and predict the relevance of target documents. (This rules out nearest neighbor approaches.)
- It must provide a prioritized ranking over the target documents, by decreasing predicted degree of responsiveness, not just a binary partitioning into "predicted responsive" and "predicted non-responsive". (This rules out nearest neighbor and decision trees.)
- It should directly provide reasonable predictions of the probability that a document is relevant. (This rules out most approaches, including naive Bayes and SVM.)
- It should produce a model that allows a degree of human interpretability; that allows, for instance, the system to highlight those parts of a document that have led to a prediction of responsiveness. (This rules out SVM.)

Of all the text classification technologies considered, logistic regression best matched these five criteria, both individually and together.

Logistic regression is an inference method widely used not just in text classification, but in medical research and other areas of statistics. It takes a large number of input variables (in the case of text classification, predominantly word occurrences and frequencies) and a single output variable (the binary class assignment of "responsive" and "non-responsive") across the training examples and fits an s-shaped or sigmoid curve that best maps the input variables to the binary output class, resulting in an equation for this curve (or more precisely, the multi-dimensional generalization of the curve). Then, when presented with a target document, the same variables (word occurrences) are extracted and plugged into this formula, resulting in a number between 0 and 1. This number expresses not just the relative strength of the prediction of the document's responsiveness, but (subject to the constraints of the model) a prediction of the probability that the document is relevant.

Figure 1 gives a slightly simplified example of single-term logistic regressions. The data set is a selection of new articles from the RCV1v2 test collection[1]. The issue we are attempting to match is those news article that are human-coded as being about industrial or corporate performance, and the evidence we use is the proportion of words in a document that are a particular term. We see that the term stem "amount" is a positive indicator of responsiveness: if the only evidence we had was that 3% of the words in an article were "amount", then we'd assign a 60% probability that the article was about corporate performance. The word stem "foreign", on the other hand, is negative evidence of relevance to this topic, whereas the word stem "austral" (for "Australia", "Australian", "Australasia", and so forth) is neutral, not providing evidence either way. (Note that in a full logistic regression, we do not fit the words individually, but instead build a single multivariate model fitting a sigmoid hyper plane to all the terms at once; additionally, the term frequency is represented in a slightly more sophisticated way.) A particularly attractive feature of logistic regression is that it assigns weights to the predictive value of individual features (in text classification, of individual terms or phrases). We can say, that is, whether the learnt model regards a certain term, or indeed a combination of terms, as positive or negative evidence of responsiveness.

This knowledge can be used at the macro level, to describe the overall sense of relevance that the model has learnt, as well as at the micro level, to highlight and direct a reviewer's attention to highly relevant terms within a document. The interpretability of the logistic regression model is a large part of its popularity within medical research: it allows an analysis of the different factors contributing (say) to recovery from a disease, even in research areas where prediction itself is not of interest. In predictive coding, the use of logistic regression gives us this interpretability for free.

[1] http://www.jmlr.org/papers/volume5/lewis04a/lyrl2004_rcv1v2_README.htm

Published research results show that logistic regression has predictive accuracy for text classification that is top or near top-of-class. And with modern, optimized tools, a logistic regression can learn from thousands of labeled documents, and make predictions on hundreds of thousands of target documents, in a matter of seconds.

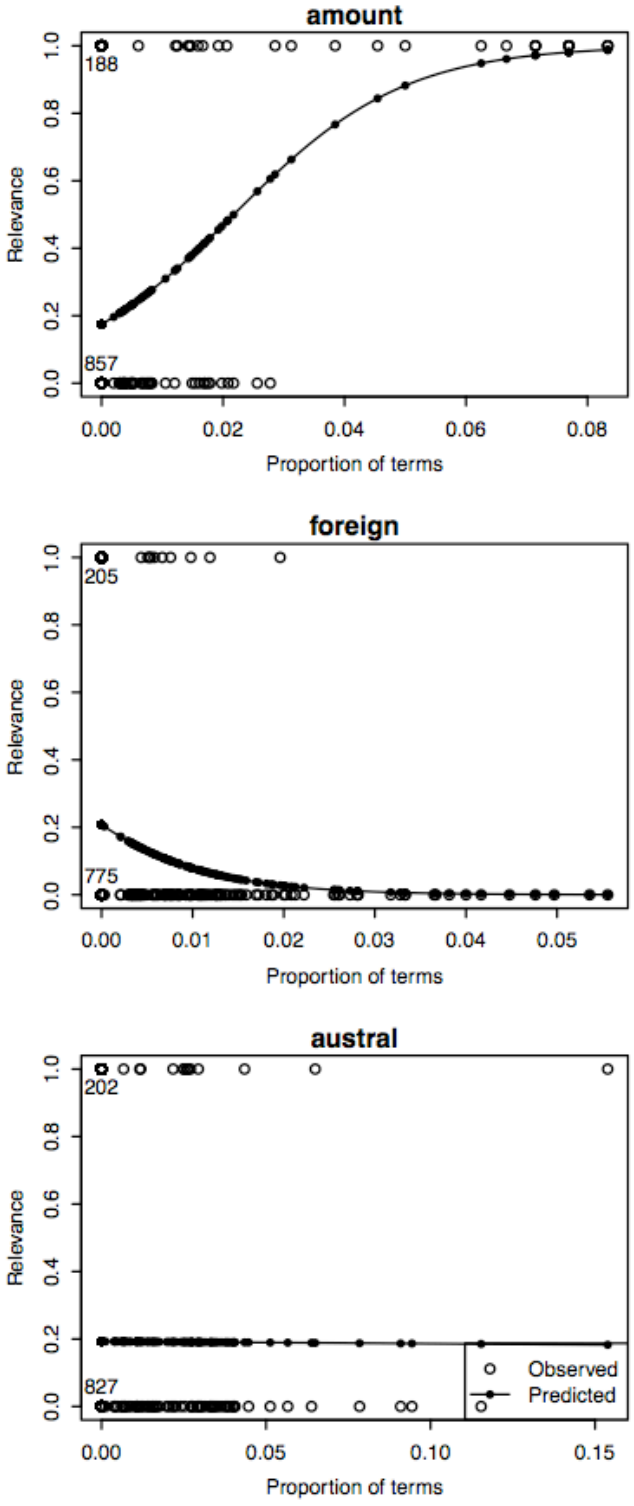


Figure 1: Term logistic regressions for RCV1v2 topic "C15: corporate performance"

3.2 Depth for recall: measuring the user's cost–benefit tradeoff

Sample-based measurement and estimation plays a very important role in predictive coding, both in guiding the review manager on when to stop training and review, and in assuring that the final production has acceptable levels of completeness and exact- ness. The most important single statistic is that of recall: that is, out of all the relevant documents in the collection, the proportion that are in the (candidate) production. Of course, a predictive coding system can achieve 100% recall by just retrieving the en- tire collection and passing it on the reviewers; but then the cost of reviewing all the documents to sift out the responsive ones will be prohibitive. The review manager is therefore faced with two key tradeoffs that he or she must decide on:

- How much review effort would be required to achieve an acceptable level of recall, if I were to stop training the predictive coding system now?

- Do I believe that further training of the system will improve it so as to provide enough savings in review cost to justify the additional cost (and delay) in training?

Unfortunately, most existing predictive coding platforms are not set up to answer these questions. Indeed, some popular ones do not even allow the review manager to make the cost–recall tradeoff at all: they produce a binary partition into predicted responsive and predicted non-responsive, and if the recall and cost level achieved by this partitioning is not suitable, there is essentially no recourse for the review manager to adjust them. Other systems do provide a predictive ranking over the collection, but do not provide adequate guidance to make the tradeoff. Still others only provide this guidance once training is completed; while training is ongoing, the review manager is only presented with an abstract measure of classifier effectiveness, with little or no information on what that means for total review cost.

In contrast, the Brainspace Discovery system places the cost-for-recall tradeoff at the center of its statistical evaluation and reporting system. From the very start of training, the review manager is informed of the depth to which review would have to be performed in order to achieve the level of recall that the manager has specified. This level of recall can be adjusted, to consider other tradeoffs. The depth for recall performance of the system is tracked over time, allowing the manager to judge not just when further training is no longer improving classification accuracy, but when actual though marginal improvement no longer justifies the training cost. And if the review manager is considering whether to cut training short even before this, due to timetable or personnel pressures, he or she knows what the review cost would be at each stage—and what savings in review cost could be achieved by lowering the recall target.

Figure 2 illustrates these differences in classifier output and reported system metrics. Figure 2(a) shows the evaluated effectiveness of a system that provides only a binary partitioning into predicted responsive and predicted non-responsive classes. The system has performed this partitioning, based upon some internal criteria, in a way that gives high precision, but completely inadequate recall. In Figure 2(b), the system has

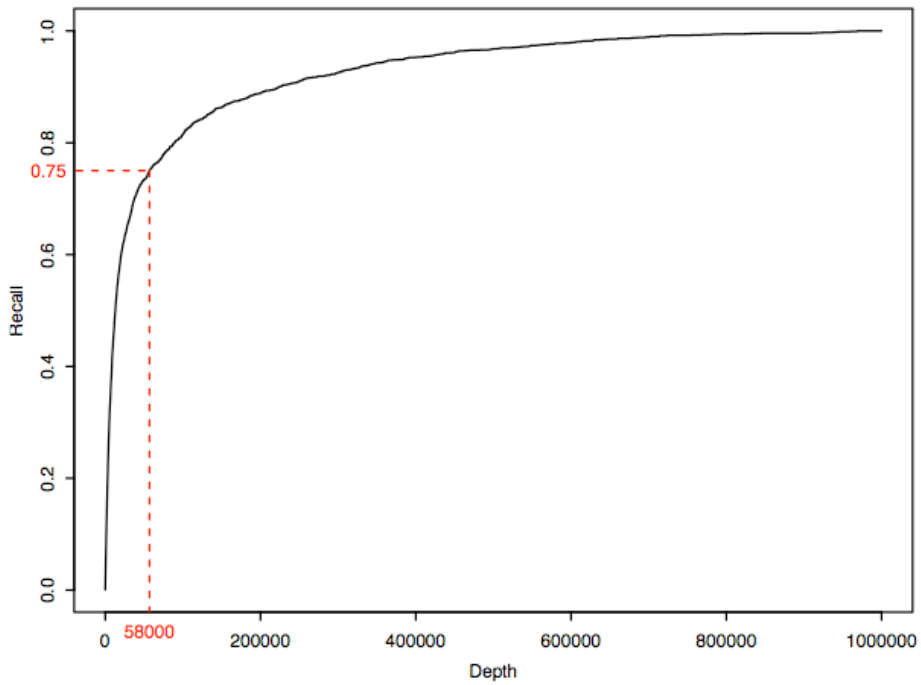
produced a ranking and then selected a cutoff point that optimizes F score. Here, recall and precision are nearly balanced, but recall is still too low, and the review manager is not guided on the cutoff that would achieve a more acceptable recall. In contrast, Figure 2(c) shows a depth for recall curve, based upon a full relevance ranking. This curve gives the estimated recall that would be achieved at each cutoff in the ranking, and therefore the number of documents that would have to be reviewed to achieve that recall. As an example, say the review manager's recall target was 75%. From this curve, we can see that this would require reviewing 58,000 of the 1 million document collection. The review manager is in control, and is no longer at the mercy of the cutoff selected by the predictive coding system.

Recall	Precision	F score
21.2%	70.5%	32.6%

(a) Recall and precision for binary production.

Recall	Precision	F score
45.0%	47.1%	46.0%

(b) Recall and precision for "optimal" F score cutoff.



(c) Depth-for-recall curve

Figure 2: Different methods of reporting system effectiveness.

3.3 Diversified active learning: spanning the information space

Training a predictive coding system is an iterative process. Partly, this is because it is difficult to know in advance how many training examples will be required; setting a fixed amount would risk having either too many or too few; therefore, we add training examples a batch at a time, and monitor effectiveness until it stops improving, or better until the improvement no longer justifies the cost (Section 2). There is, however, another compelling reason for selecting our training examples in an iterative manner, and that is that a classifier built upon a partial set of training examples can be very helpful in suggesting useful future training examples for labeling.

The process of using the (partially-trained) classifier to select additional training examples is known as active learning, and it can lead to a dramatic speedup in training (and hence saving in both cost and elapsed time). A central idea in active learning is that there is little value in labeling training examples that the classifier is pretty certain are either responsive or non-responsive. Rather, training effort should be focused on those documents that the classifier is unsure about, and which once labeled are likely to change and improve the classifier's model most rapidly. This is particularly important in electronic discovery, where a large part of the collection is often made up of clearly non-responsive or largely repetitive material: repeatedly including these documents in the iterative training rounds when the classifier already knows how to classify them is clearly a waste of time.

While many predictive coding systems have yet to implement active learning, its importance in saving training time, effort, and cost is being increasingly appreciated. Many of the existing implementations, however, only consider uncertainty in selecting documents for training: that is, their only criterion for selecting documents is to choose those that are as likely to be relevant as to be irrelevant. While this is an important criterion (and some popular active learning systems omit it to their peril), there are two other important criteria to be considered:

- Diversity: is this document similar to others that we have already trained—or that we have already selected for this training batch?

- Density: is this document similar or dissimilar to other unlabeled documents? That is, will it help us decide the responsiveness of many other documents, or only of itself?

Diversification is an important consideration in any setup, but particularly so given that in predictive coding, training documents are typically selected and labeled in batches of as many as a few hundred at a time. Much of the research literature assumes, in contrast, that training documents are chosen one document at a time; but this is generally impractical in predictive coding. The problem with using only uncertainty to select a batch of training examples is that all the documents whose responsiveness predictions happen to be the closest to 50:50 on a given iteration, will also tend to be similar to each other. If you've used a

competitor's system and found that at each training round you were getting many documents that were like each other (and which were, moreover, poorly representative of the collection as a whole), then you have encountered this problem.

Brainspace Discovery, in contrast, has all three strands of criteria—uncertainty, diversity, and density—built into its selection of training examples. At each training round, the trainer sees not simply those documents that the classifier is most uncertain about, but uncertain documents that are diversified across the collection, and provide insight into concentrations of interesting material. As a result, the information space of the collection is spanned by the training examples more rapidly, leading to faster learning—and better understanding by the trainer and review manager of what the col- lection holds.